Manual:LocalSettings.php

Warning: Don't edit LocalSettings.php with Notepad, TextEdit, or any other text editor that adds byte order marks to files, or you will break your wiki. Instead, use editors like Notepad++ or Vim; they handle file encodings properly. Notepad++ and Vim can also be used to remove an erroneously added byte order mark from the beginning of the file again.

MediaWiki File: LocalSettings.php	
Location:	/
Source code:	Not in Git (created during installation)
Classes:	None

The `LocalSettings.php` file provides basic configuration settings (based on the `DefaultSettings.php` file) of a MediaWiki installation. You should take your time to review the settings in this file. The file is usually generated by the web-based MediaWiki installer but you can tweak the parameters, possibly in concert with Apache settings. For MediaWiki installations configured as a wiki farm, a file named `CommonSettings.php` may be used.

Contents

1 Location on the server
2 Security
3 File contents
4 Overview of available settings
5 Standard settings
5.1 Include path
5.2 Site name
5.3 Declensions of site name
5.4 Site language
5.5 Script path
5.6 Server name
5.7 Script name
5.8 Redirect script name
5.9 Article path
5.10 Stylesheet location
5.11 Upload location
5.12 Logo
5.13 Contact info
5.14 Database settings
5.15 User rights
5.16 Force capital links
5.17 Enabling subpages
5.18 Image uploads
5.19 LaTeX inline equations
5.20 Interwiki support
5.21 Language of user interface
5.22 Setting copyright for the site
5.23 Custom namespaces
5.24 Skins
6 See also

Location on the server

The LocalSettings.php file is not a wiki page and you cannot access it with your web browser. Instead, it is a file in the file system of the server. Its contents are generated during the initial setup of the wiki and the resulting file must be copied on the server manually. The file must be located in the folder you installed MediaWiki into, on the same level with folders like "includes/" or "skins/" and files like "api.php". If this file is *not* there, the wiki will not work at all - if the wiki *does* work, the file *is there*. If you do not know, where it is, you can run this command `find / -iname 'LocalSettings.php' -print` on the UNIX shell to make your server show you, where the file actually is.

Security

LocalSettings.php usually contains sensitive data such as database logins. This data should *never* be revealed to the public! Due to a security breach somewhere on the server, it might happen that other users are able to view the contents of files. In order to improve security of your data, you should set UNIX permissions for this file accordingly: The webserver user must have access to this file. If this is the same account, who is the owner of the file, then you can set permissions to *600*. Sometimes, the webserver user is *not* the file owner, but he is in the owner's UNIX user group. In this case, permissions of *640* should be fine. For improved security you should narrow permissions down as far as possible.

Additionally, you can create a MySQL user, who is restricted to only the database used by the wiki and provide this user's credentials in LocalSettings.php. Also can you configure your database server to only accept connections from localhost - this should prevent access from outside in case of leaked credentials.

File contents

When you edit LocalSettings.php, make sure to save it in the right encoding again. You should use "ANSI as UTF-8" encoding - that is UTF-8 encoding without byte order mark (BOM).

LocalSettings.php contains PHP code; mainly the definition of variables and their values. Changing a setting usually means changing the value of a PHP variable. Your changes will take effect immediately after saving the file again on the server: There is no need to manually "restart" anything. Anyway, in some cases you may need to clear your browser's cache to actually see the changes you made.

The *default* values of many more settings are set in `includes/DefaultSettings.php`, *which should not be edited;* if the variable you want to change is not already mentioned in your `LocalSettings.php`, copy the appropriate line from `DefaultSettings` and modify it appropriately in LocalSettings.php. Within `LocalSettings.php` you can add new lines at the end.

Within the file, there can also be several lines such as '`require_once "extensions/extension.php";`', which point to extensions. These lines enable the according extension in the wiki. Those extensions may require setting the values of more variables in `LocalSettings.php`; check the documentation of the according extension for further instructions.

> In **MediaWiki 1.16 and earlier** it was important to not place extensions before the `require_once "includes/DefaultSettings.php";` line; that will blank the extension setup function arrays, causing no extensions to be installed, and probably making your wiki inaccessible. Since 1.17 it's no longer necessary for LocalSettings.php to include DefaultSettings.php.

Your version of `LocalSettings.php` may not end with a closing PHP tag of `?>`. This is done on purpose, as it prevents people from accidentally adding new information *after* this tag. PHP will function just fine without a closing tag.

If you have a wiki on a MediaWiki wiki farm you may not have write-access (perhaps not even read-access) to the file LocalSettings.php (see e.g.[1]). The wiki farm company may or may not be willing to make changes you desire. Perhaps it wants to keep most settings the same on all wikis of the wiki farm.

> Before 1.17, this file was automatically generated by the installer script and placed in the 'config' folder. As of 1.17, this file is no longer written to your webserver for security reasons. It is now offered as a download, which you must then upload to your server for your wiki to begin working. For a more complete explanation, see this explanation .

Overview of available settings

See the configuration settings index and the comments included in the settings files for help on what all the variables do. A short listing of the most important variables, as well as the most requested features, is listed below.

Standard settings

Include path

The **$IP** (include path) variable holds the local file path to the base installation of your wiki. Make sure you refer to the fully qualified file path. Do not include a slash at the end of the path. Since MediaWiki 1.17, setting $IP in LocalSettings.php is no longer needed. It will default to the current working directory automatically.

The DefaultSettings.php file gets loaded from the directory designated by the IP variable.

Site name

The **$wgSitename** variable holds the name of your wiki setup. This name gets included many times throughout the system, such as via MediaWiki:Pagetitle. For instance, the Wikipedia tagline "'From Wikipedia, the free encyclopedia."' makes use of this setting.

Declensions of site name

Some translations of interface are ready for inflection of site name. You can set proper forms of word in variables **$wgGrammarForms**.

Site language

The **$wgLanguageCode** variable controls the language of your wiki's interface. While users can switch the language they see in their preferences, this variable sets the default language that all anonymous users and most registered users see.

Script path

$wgScriptPath is the URL path prefix to access the main MediaWiki script that is the central acting piece of code of MediaWiki. This setting should correspond to the Apache settings, especially if you are using Apache's rewrite rules.

Server name

$wgServer can be used optionally to overwrite the automatic detection. When the wiki is accessed from either the localhost where it runs, from an intranet or from the internet, e-mail notifications and a few other computed messages would be usually delivered with the different URLs. Using `$wgServer` overwrites the detection and the server name will be constant.

If www.example.com is the server address of your wiki as seen from the internet, add a line such as

```
$wgServer = 'http://www.example.com';
```

The magic word variable `{{SERVER}}` can be used on wiki pages; it equals the value of `$wgServer`; however, on a wiki, such as those in the WikiMedia family, where relative urls are used, it will not fully expand, for example here it displays //www.mediawiki.org, which can neither be clicked nor copy-pasted into the address bar as it is.

Script name

$wgScript is the name of the main (index) MediaWiki PHP script, named `index.php` by default. Changing the script name is probably not a good idea. However, if you feel the urge to do so, this is the place to make your changes. Make sure you know what you are doing.

Redirect script name

The **$wgRedirectScript** variable specifies the URL path to the *redirect script* which handles redirects in MediaWiki. This setting is similar to wgScript as you can configure the exact name of the script. As mentioned above, this is probably not a good idea unless you really know what you are doing.

Article path

$wgArticlePath is the path to use when accessing a page in MediaWiki. The path should contain the path to the main script (usually making use of `$wgScript`) and use the `$1` placeholder for the article name.

If you are using Apache rewrite rules to create pretty and short URLs, you probably need to adjust `$wgArticlePath` to address the right path. Note that wgArticlePath is used to construct URLs from within MediaWiki. If you make a mistake here, internal links will show up incorrectly while you still may be able to access the main page by specifying the correct URL manually. See Manual:Short URL for more information on URL configuration.

Typical values are:

"`$wgScript/$1`"	pass the article name with separator "/"
"`$wgScript?title=$1`"	pass the article name as a parameter (old style)
"`/mypath/$1`"	custom path. Use Apache rewrite rules to convert "mypath" to the proper path accessing the main script

Stylesheet location

Use the **$wgStylePath** variable to set the URL path to the place where the stylesheets (CSS) for the MediaWiki installation are located. The **$wgStyleDirectory** variable should point the same place, but note that this is a local file system path for use in internal scripts accessing the file system.

Upload location

The upload directory is the place where files uploaded by the users are stored. The **$wgUploadPath** variable specifies the URL path, the **$wgUploadDirectory** points to the local file system path.

Logo

The **$wgLogo** variable specifies which graphical logo is displayed in the top left corner of all mediawiki pages. These steps replace the default logo in the /wiki/skins/common/images/ directory, replacing /wiki with the path to the directory where you installed the MediaWiki software.

First, copy whatever logo you want into the /wiki/skins/common/images/ directory (replacing /wiki with the path to the directory where you installed the MediaWiki software).

Next, try to find the line that looks like this in `LocalSettings.php`:

```
$wgLogo         = "{$wgStylePath}/common/images/wiki.png";
```

If there's no such line, you can copy and paste the line above at the end of the file.

Then, modify that line to point to your logo. The logo must be web accessible. The value of this variable is passed to the web browser, which uses it to fetch the logo. If in doubt, a good way to pick what to put here is navigate to the logo in your web browser (for example, this wiki the url of the logo is http://upload.wikimedia.org/wikipedia/mediawiki/b/bc/Wiki.png), and put the full url as the value of this variable.

Some people just replace the skins/common/images/wiki.png file with their logo. This is not recommended, as the customized logo would be overwritten on upgrade.

Contact info

The **$wgEmergencyContact** variable is the e-mail address of the user to contact if something goes wrong. This e-mail address is used to send internal bug reports to. As an administrator, you want to include your e-mail address here.

The **$wgPasswordSender** variable is the e-mail address where e-mail gets sent *from*, when passwords are sent out to users who have forgotten their passwords. Choose an address people can reply to in case of trouble or confusion.

Database settings

MediaWiki needs access to the database (currently either MySQL or PostgreSQL) to store pages, modifications, user information, and a lot more things.

The **$wgDBserver** contains the hostname where the database is hosted on. In most cases this will be just "localhost"

as the database is run on the same system, but for distributed installations, you need to fill in the fully qualified domain name of the computer running the database.

$wgDBname is the *database name* of the database to be used by MediaWiki. A single MySQL or PostgreSQL installation can store more than one database and you can even run many MediaWiki installations on a single server. Make sure you have stated the correct database name here and use different database names for different wiki installations on the same database server.

The **$wgDBuser** and **$wgDBpassword** variables contain the login name and password to be used by MediaWiki to access the database. Make sure the specified user has the proper access rights to be able to manipulate the wiki's table on the database server.

User rights

The **$wgGroupPermissions** is an associative array, controlling permissions for creating and editing pages for your different user groups. In this array, custom permission levels can be created, and permission levels for the different user groups can be set. See Help:User rights for more information about the different permissions and user groups available.

Force capital links

By default, no page name can start with a lowercase Roman letter: in an attempt to do so the first letter is converted to uppercase; if a link target, included page, image or category is specified with a name starting with a lowercase letter, the actual target etc. is the page starting with the corresponding capital.

Alternatively page names *can* start with a lowercase letter, in addition to the possibilities of starting with a capital, a digit, etc. For that you must adjust the **$wgCapitalLinks** variable. Setting it to **false** allows lowercase characters, **true** chooses the default behaviour.

Enabling subpages

On localsettings.php, subpages are enabled on a per-namespace basis using the **$wgNamespacesWithSubpages** variable. For example, to enable subpages in the main namespace:

```
$wgNamespacesWithSubpages[NS_MAIN] = 1;
```

Image uploads

Before users are allowed to upload files to the MediaWiki system, you have to enable that feature. Make sure the Upload Directory is properly configured and writeable by the Apache web server process. Then set the **$wgEnableUploads** variable to **true** to allow uploading in the web user interface.

i.e. Here's some example code from *includes/DefaultSettings.php* to put in *LocalSettings.php*.

```
$wgUploadPath       = "$wgScriptPath/uploads";      ## Wiki 1.5 defaults to /images, but allows more than just images
$wgUploadDirectory  = "$IP/uploads";                ## Wiki 1.5 defaults to /images, but allows more than just images

## To enable image uploads, make sure the above '$wgUploadPath' directory is writable by Apache User or group.
## ''(i.e.  chmod og+w uploads images)''  then the following should be true:
$wgEnableUploads       = true;

$wgUseImageResize      = true;
$wgUseImageMagick      = true;
$wgImageMagickConvertCommand = "/usr/bin/convert";

## If you want to use image uploads under safe mode, create the directories images/archive, images/thumb and
## images/temp, and make them all writable. Then uncomment this, if it's not already uncommented:
$wgHashedUploadDirectory = false;
```

If you want to be able to resize images on the fly to support thumbnails, MediaWiki needs a working ImageMagick installation. Set the **$wgUseImageResize** and **$wgUseImageMagick** variables to **true** once you have installed and tested ImageMagick on your system. Make sure the **$wgImageMagickConvertCommand** variable points to the proper location of the `convert` command of your installation, that the command is executable by the web server process, and **$wgMaxShellMemory** is large enough. See Manual:Image Administration#Image thumbnailing for detailed information and trouble-shooting.

Also, you may want to modify the list of accepted extensions, which is stored within the **$wgFileExtensions** array:

```
$wgFileExtensions = array('png','jpg','jpeg','ogg','doc','xls','ppt','mp3','sxc','pdf');
```

In case you run into trouble with ImageMagick, and obtain the following error message:

`Fatal error: mime_magic could not be initialized, magic file is not available in includes/MimeMagic.php on line 506`

Try adding the following line to LocalSettings.php:

```
$wgMimeDetectorCommand = "file -bi";
```

Read the upload security section in the Manual:Security.
Further information is available in Manual:Configuring file uploads, Manual:Mime type detection, and Manual:Image Administration#Image thumbnailing.

LaTeX inline equations

In order to render mathematical formulae in LaTeX, you need a working LaTeX installation on your system. Set the **$wgUseTeX** variable to **true** to enable that feature. Also make sure the **$wgMathPath**, **$wgMathDirectory** and

$wgTmpDirectory are set properly, but you might want to go with the default values unless you have a very good reason to change it.

Interwiki support

InterWiki support is built into MediaWiki but you need to configure the prefix to be used for your internal links. This prefix is usually the same as $wgSitename, but in case you need to change that, you set the **$wgLocalInterwiki** variable to the preferred name.

Language of user interface

MediaWiki allows for a variety of localized user interfaces languages instead of the english default. If you want to run your wiki in a non-English language, set the **$wgLanguageCode** variable to the proper language code (e.g. "de" for German, "es" for Spanish, etc.)

You may use any of the languages found in the directory language. E.g., if you find MessagesEs.php, you can use "`es`" for Spanish. Do not use "`Es`" with a capital letter. Although it seems to work, not all texts are translated.

Note: After changing the language code, you will need to run a php script to make it work. With your command line tool, go to your wiki directory, open the "maintenance" folder, and type "`php rebuildMessages.php --rebuild`". Please note that on Windows, the PHP folder might not be included in the PATH environment variable.

Not all languages are supported. See here for a list of the localisation statistics of 1.24wmf3 (7d9e709). See translatewiki.net for the statistics per released version.

Setting copyright for the site

- **$wgRightsPage** is the page on the wiki that covers the copyrights that it falls under. Usually, this will be Project:Copyrights.
- **$wgRightsUrl** is the page describing full details of your license. (For the GNU FDL, for example, this would be `http://www.gnu.org/licenses/fdl.html`.)
- **$wgRightsText** is the text in the footer that follows "Content is available under". It will be linked to the page specified in `$wgRightsPage`.
- **$wgRightsIcon** is the URL of the image placed at the left of the footer.

  Note: If `$wgRightsPage` is non-empty, the link in the copyright/license notice will link to that page on your site. If `$wgRightsPage` is empty then the copyright/license notice will link to `$wgRightsUrl` instead.

To modify the copyright statements of the site, add something like this to LocalSettings.php:

```
$wgRightsPage = "YourWiki:Copyright";
$wgRightsText = "copyright YourWiki";
```

Afterwards, edit MediaWiki:Copyright to provide an appropriate message, using `"$1"` to indicate the position where the link to your copyright page will be.

Example: setting a Creative Commons license

To set a Creative Commons license do the following:

- Choose your license: http://creativecommons.org/text/publish-website
- Examine the returned HTML code, e.g.:

```
<a rel="license" href="http://creativecommons.org/licenses/by-nc-sa/3.0/">
<img alt="Creative Commons License" border="0" src="http://creativecommons.org/images/public/somerights30.png" />
</a>
This work is licensed under a
<a rel="license" href="http://creativecommons.org/licenses/by-nc-sa/3.0/">
Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License</a>.
```

- For `$wgRightsURL` enter the href info from the first anchor:

```
$wgRightsUrl = "http://creativecommons.org/licenses/by-nc-sa/3.0/";
```

- For `$wgRightsText` enter the text the second anchor links from (add the "a" to the Wiki variable for a more grammatically correct version)

```
$wgRightsText = "a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License";
```

- For `$wgRightsIcon` enter the src info from the img link in the first anchor. You want to copy the badge to your own site instead of using the one from the Creative Commons site.

```
$wgRightsIcon = "http://creativecommons.org/images/public/somerights30.png";
```

- For using RDF metadata about copyright add one or both of the following lines to your LocalSettings.php

```
$wgEnableCreativeCommonsRdf = true;
$wgEnableDublinCoreRdf = true;
```

Custom namespaces

By declaring the **$wgExtraNamespaces** array, and modifying the **$wgNamespacesWithSubpages**, and **$wgNamespacesToBeSearchedDefault** arrays, extra namespaces can be added to a MediaWiki installation; and by declaring the **$wgNamespaceAliases** array namespace aliases can be added. **Take heed** not to have any pages already titled in that namespace, for instance in you had a page called "Technical:Support" and you created the Technical: namespace, then that page would not only be lost, but you cannot remove it from Special:Allpages. To fix this delete the namespace, move "Technical:Support" to "Support" in mainspace, delete the redirect, reinsert the namespace, and move it back to "Technical:Support". See Custom namespaces for more information as to how to do so.

Skins

The default skin of the site can be modified via the **$wgDefaultSkin** variable.

See also

- Wiki families can share the same LocalSettings.php.
- Extension:Configure provides a way to set configuration settings from a special page.
- Manual:Securing_database_passwords - a guide to securing your database passwords and LocalSettings.php file.

Language:	**English** • Deutsch • Ελληνικά • español • français • Bahasa Indonesia • 日本語 • 한국어 • português do Brasil • русский

Categories: MediaWiki code | MediaWiki configuration

This page was last modified on 3 May 2014, at 19:34.

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. See Terms of Use for details.

Privacy policy About MediaWiki.org Disclaimers Developers Mobile view